REMARKS

Applicants have made several corrections to typographical errors. No new matter has been introduced by these changes.

The Patent Examiner rejected claims 1-12, 14-18, 20-23, 28-30, and 32-34 as anticipated by US Patent No 6263498 ("Alcorn"). Applicants respectfully traverse.

Anticipation under 35 U.S.C. §102 means lack of novelty, and is a question of fact. To anticipate, every element and limitation of the claimed invention must be found in a single prior art reference, arranged as in the claim. *Karsten Mfg. Corp. v. Cleveland Golf Co.*, 242 F.3d 1376, 1383 (Fed. Cir. 2001); *Scripps Clinic & Research Foundation v. Genentech, Inc.*, 927 F.2d 1565, 1576 (Fed. Cir. 1991).

As described on pages 3 and 4 and summarized on page 5 of the present application, a "tier" is "A logical grouping of components that perform a well-defined, predetermined function." A "framework" is "An architected context for business objects that modify the business objects' attributes or add new behavior." An "object" is "A programming structure encapsulating both data and functionality that are defined and allocated as a single unit and for which the only public access is through the programming structure's interfaces. A COM object must support, at a minimum, the IUnknown interface, which maintains the object's existence while it is being used and provides access to the object's other interfaces." An "n-tier architecture" therefore "describes an architecture that is characterized by a plurality of 'N' tiers 30, each of which has a specified type and a specified interface. Although a hierarchy can be defined for the tiers, no hierarchy is implicit in the N-tier architecture of the present invention."

Alcorn discloses a method for managing application modification in a distributed data processing system. Col 2 ll. 11-27. As such, it discloses modifications to an instance of an

object where the instance adopts a modification. *Id.* Moreover, Alcorn discloses use of so-called JAVA beans and dips. A "dip" is a special kind of JAVA bean that can be hooked on to another JAVA bean. The dip is the new feature that one wants to add to the component. Software examples of dips include printing and security

Moreover, Alcorn uses "tier" in a different manner altogether from that which is used in the present invention:

> Many types of programs are presently available in both a stand alone form and from a server in a network that employ various rules for decision making. In addition, some programs are executed in a "multi-tier" distributed environment in which applications may use multiple computers to solve specific problems. It is often desirable to modify only the portion of the application on a server as opposed to modifying the portion of the application on a client machine.

Alcorn, at best, only describes a two-tiered architecture as that term is commonly understood by those or ordinary skill in this art: "In a multi-tier distributed environment, distributed applications are provided in which a portion of the application is located on a server, such as server 104 and another portion of the application is located on a client, such as client 108. In this implementation, the client is considered a first tier system while the server is considered a second tier system."

Specifically, Alcorn does not disclose a software architecture as the term is used in the present invention. Moreover, Alcorn does not disclose building N-tier software applications. Alcorn's disclosure that CORBA specifies an extensive set of bus-related services for creating and deleting objects, accessing them by name, storing them in persistent stores, externalizing their states, and defining ad-hoc relationships between them is not the same as specifying component rules.

Alcorn, never disclosing tier rules as used in the present invention, further fails to disclose specifying a set of tier rules for creating an extensible set of tiers. Alcorn merely discloses DAP server bean interfaces and DAP client beans. These are both objects, not tiers.

The Examiner's assertion that disclosure of objects in a dipping process is similarly flawed as this object behavior is not a set of association rules by which at least one software component created using the software component rules may be associated with or disassociated from at least one tier created with the set of tier rules. In fact, there is no association rule at all cited within Alcorn.

DIPS are not a set of tier framework rules, and Alcorn teaches away from the present invention by noting that predefined classes are typically called "frameworks" and that JAVA supplies the framework. In fact, the rules disclosed by Alcorn are those which manipuilate a "dip," which is a class that has an interface that allows the class to be used by a dippable object. Alcorn's dippable object 504 looks and behaves like object 502, but has the added ability to accept behavioral modification components. This is far from providing an architected context for software components associated with a tier. Additionally, Alcorn does not disclose a set of package rules to provide for logical grouping of interfaces within a framework defined by the tier framework rules to provide a set of specific behaviors for the tier, as it teaches that the frameworks are provided by JAVA.

Finally, Alcorn does not address specifying a set of assembly rules, the assembly rules comprising association rules by which each tier may be associated with at least one other tier and linkage rules by which each tier may be linked to at least one other tier. Assembly rules comprise rules on methods of assembly of tiers 30 and software components 20 into final

applications. That on which the Examiner relies in Alcorn is merely an application programming interface.

The Patent Examiner's reference to Alcorn discloses CORBA services and business rules. Services are methods exposed by an object to other objects. Business rules, as used by Alcorn, are understood to mean means by which methods may be applied to properties and/or other data contained within, exposed to, or sent to the object. A business rule, accordingly, is not a rule for creating an extensible set of tiers. Further, the section of Alcorn to which the Patent Examiner refers is absolutely silent on associating an object with a tier. To the contrary, the process disclosed in Alcorn creates an instance of an object, presumably within whichever tier (if any) the parent object resides. Additionally, Alcorn fails to disclose any rules regarding frameworks or their rules or packaging rules as taught and claimed by the present invention. Customization of the JDBC link to the server or between two dips is not an assembly rule, but is rather another instance of modification of a property used to help implement a service. Therefore, Alcorn fails to disclose every element and limitation of the claimed invention and certainly fails to disclose the elements and limitations as arranged as in claim 1.

Accordingly, Alcorn does not disclose an extensible, n-tiered architecture, much less a system for accomplishing such an architecture.

It is axiomatic that the remainder of the claims depending from claim 1 are likewise not anticipated by Alcorn as claim 1 is not anticipated by Alcorn. However, certain limitations and assertions by the Examiner also require additional traversal.

Alcorn discloses neither an inventory of objects nor placing objects into or removing them from an inventory.

Alcorn does not disclose use of an iterator class (which is not the same as a dip or dippable object).

Accordingly, Alcorn fails to disclose every element and limitation of the claimed invention arranged as in the claim. Alcorn, as a matter of law, therefore does not anticipate the present invention under 35 U.S.C. §102.

The Patent Examiner rejected claims 13, 19, and 31 as being obvious under 35 U.S.C. 103(a) in view of combining Alcorn with US Patent 6446113 ("Ozzie").[1] Applicants respectfully admit that asynchronous and synchronous communications are well known in the art. However, as traversed above, Applicants respectfully submit that the prior art does not disclose that which is claimed in claim 1 and that these claims, depending from allowable claims, are themselves allowable.

Regarding claim 19, Applicants respectfully submit that the sentence in the Office Action is unintelligible and Applicants cannot therefore traverse. Clarification of this rejection is therefore respectfully requested.

The Patent Examiner rejected claims 24 and 27 as obvious under 35 U.S.C. 103 in view of combining Alcorn with US Patent No. 6263498 ("Gish"). Applicants respectfully traverse.

As traversed above, Alcorn does not disclose the present invention. As claim 1 is patentably novel and unanticipated, claims 24 and 27 are unanticipated, depending from an allowable claim. Applicants acknowledge that testing software is well known in the art. Neither reference cited by the Patent Examiner, however, teaches an inventory for an n-tier extensible architecture as claimed in claims 24 and 27.

---

[1] The Patent Examiner refers to this reference as "Ozze," but presumably meant "Ozzie."

It is submitted that claims 1-34 are now all in condition for allowance.

Respectfully submitted,

October 31, 2003

Gary R. Maze
Reg. No. 42,851
Duane Morris LLP
One Greenway Plaza, Suite 500
Houston, TX 77046
Tel.: 713.964.7700
Fax: 713.964.7701

CERTIFICATE OF MAILING 37 CFR 1.8(a)

I hereby certify that a copy of this document along with any referred to as attached or enclosed is being deposited with the United States Postal Service as First Class mail, postage prepaid in an envelope addressed to Mail Stop Non-Fee Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on October 31, 2003.